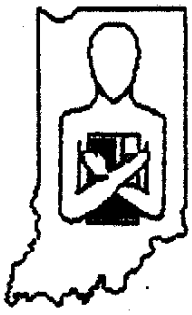


HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP
USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER
GROUP HOOSIER USERS THE HUGgers GROUP HOOSIER USERS
HOOSIER USERS GROUP HOOSIER USERS GROUP
USERS GROUP HOOSIER People Helping People USERS GROUP HOOSIER
GROUP HOOSIER USERS GROUP HOOSIER USERS
HOOSIER USERS GROUP HOOSIER USERS GROUP
USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER
GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS

FEBRUARY, 1986 THE HUGgers NEWSLETTER VOLUME 3, NUMBER 11

THE OFFICER'S CORNER

Well, let's get the important stuff out of the way first.

Due to the upcoming Indiana Hamfest on March 9, our March Monthly Meeting will be postponed one week. The March Meeting will be held on March 16th, same time, same place.....

Also, while on the subject of computer related fests', the LA99ers will be hosting a 99/4A Expo in Los Angeles next month. (More info and a discount coupon at the right.) In recent correspondence received from Millers Graphics, they say "If all goes well we will be announcing a NEW piece of hardware and showing some new software at this Fest!" I only have one question, what could Craig and Company come up with that tops the Gram Kracker?

Back on the home front..... It's election time again. We'll need a few able minded volunteers to put their heads together and come up with a slate of Officers to be presented at the April, 1986 meeting. I, myself, do not plan to actively pursue the office as President of our group. If you would like to be a part of the Nominating Committee, see me at this meeting.

As you can see, I'll be giving a workshop on the Gram Kracker again this month. Now that the holidays are past and things have settled down, I've had more time to tinker with it.

Thats it for this month. See you at the February 9th meeting. J.Steven Sims

THE NEXT MONTHLY MEETING WILL BE
FEBRUARY 9, 1986
STARTING AT 2:00 PM AT CREATIVE LOGIC.
SEE YOU THERE!

WORKSHOPS: THE GRAM KRACKER, PART II
PROGRAMMERS CORNER

SOUTH REGIONAL MEETING

Saturday, February 15
1:30 to 4:30 pm
(See page 2 for more details)

MARCH NEWSLETTER DEADLINE

FEBRUARY 21, 1986

EDITOR'S NOTE: For those who would like to help in publishing the April Newsletter, see Pam Sims at this meeting!

99'FEST-WEST'86
Hosted by
The Los Angeles 99er Computer Group
MARCH 1 and 2, 1986
Mezzanine Level
SHRINE EXPOSITION HALL
700 WEST 32nd STREET
LOS ANGELES, CALIFORNIA

ADVANCE ADMISSION $4.00 for BOTH DAYS,
$5 AT DOOR WITH COUPON BELOW
For more information contact: T. A. Masters, 148 S. Maple Drive, Beverly Hills, CA 90212. PLEASE INCLUDE A STAMPED SELF-ADDRESSED ENVELOPE

```
-----------------------------
!99'FEST-WEST'86 DISCOUNT COUPON!
!SAVE $1  - DUPLICATION ACCEPTED!
!        ONE PER PERSON        !
-----------------------------
```

HAPPY BIRTHDAY!

A Happy HUGger Birthday to these members who joined, or renewed in February, 1985: Vern Hines, Ken Burrell, Earl Pauley, Thomas Walker, Frank & Mid Chase, Russell Pitz, Cptn. Donald Taylor, Marijane Smith, Rene' Torella, Bill Godby, Paul & Mary Vollenweider, Bob Slomka, Daniel Poe and Dennis Graves. Its time to renew!

WELCOME!

The Hoosier Users Group would like to welcome Roy Kidd, and Terry Readman, who joined in the past month.

WELCOME BACK!

We'd also like to welcome back these renewing HUGgers: Bill Jones of Indy, Leonard & Mary Eubank, Barb Uhrig, Carl Walters, Mike Oeth, Bob Sobek, Lester Goss, and Sam Hatcher!

SOUTH REGIONAL NEWS

February's South Regional meeting will be moved to SATURDAY, Feb. 15, from 1:30 to 4:30 P.M. (You may arrive and leave any time during that time period if you have other commitments.) The meeting location is 4582 Moccasin Pl., Greenwood. You may obtain directions by calling 881-5918. Come and share programs and information with other members of your group. A copy of the club's library is available for your use.

LIBRARY BITS
by Dennis Sherfy

BASIC-3 contains a version of the game MASTERMIND. It is called CODE-BREAK. This is a "brain game" for those of you who enjoy a little mental exercise.

The principal behind the game is simple. A secret code is selected, and you must guess the code. There are two options. Eithe you select the secret code for a companion to guess, or the computer will select a secret code for you to guess.

The code is a combination of eight different colored blocks. They may be arranged in any order, and the same colored block may be used more than once in a code. You must guess which colored blocks are in the code, and their correct order. Your guess is entered by using numbers 1 to 8

There are some weaknesses in the program. There are no instructions, and you are not told which colors correspond to which numbers. If you use a monochrome monitor, you loose all distinction between the block colors. One solution is to ignore the colored blocks, and use pencil and paper. Write the numbers for each guess. This has another advantage in that you can mark your paper. I cross off numbers that I determine are not in the code, underline numbers that I know are in the code, but I'm not sure if they are in the correct position, and circle numbers that are in the code, AND in the correct position.

You may choose a code of one to nine items or blocks. The program displays colored blocks as you enter your guess, counts how many guesses you take, and gives you clues after each guess.

The clues are white or black blocks. Each white block tells you that you have guessed one item in the code, but it is not in the right position. Each black clock tells you that you have guessed one item in the code which IS in the correct position. What makes the game interesting is that the clues don't tell you WHICH item is correct. When you have guessed the correct code, it is displayed at the top of the screen. You must press ENTER to play another game, or FCTN 4 to end the program.

If you want to make a change before you have completed entering your entire guess, you may use the back arrow, FCTN S.

The game demonstrates some interesting programming. The title screen is developed by moving letters into position from the top or bottom of the screen. Keyboard entries are made similar to the Extended Basic ACCEPT AT statement. The screen is set up with Basic's equivalent of the DISPLAY AT statement in Extended Basic.

The Extended Basic manual includes a code guessing game to demonstrate many of the new statements in Extended Basic, but each number in the code can be used only once. When repetition is allowed, complexity increases greatly.

Foundation Computing out of business

Foundation Computing has gone out of business, and Kathy Hunter, vice president of marketing, described the firm as "financially bankrupt."

Foundation manufactured the Z80A card and 80-column card for the TI99/4A, as well as 32K and 128K memory cards for the machine.

Hunter described the firm's leaving business as resulting from a combination of events. The company which manufactured Foundation's boards went bankrupt "with the boards in tow," she said. She also noted that components supplied from a major company to Foundation "were faulty."

Hunter added that "John Koloen's articles were not very helpful."

Koloen, publisher of *MICROpendium*, wrote in the June 1985 issue that the Z80 card would not format a disk using the MRS operating system. In a subsequent issue he wrote that Foundation replaced his disk with one which worked properly.

Hunter said that sales of the Z80 card dropped by 75 percent following the initial article.

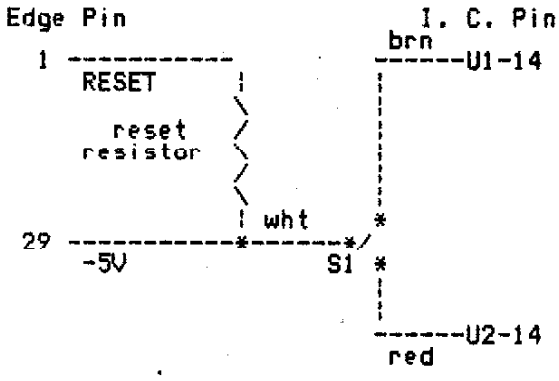
She said that the company will continue to offer maintenance and repairs for persons wishing support. For products under warranty, the customer will pay only the shipping and for products not under warranty, the customer will be charged $35 plus shipping.

New address for service is P.O. Box 455, Mill Valley CA 94942.

MORE FOR YOUR MONEY

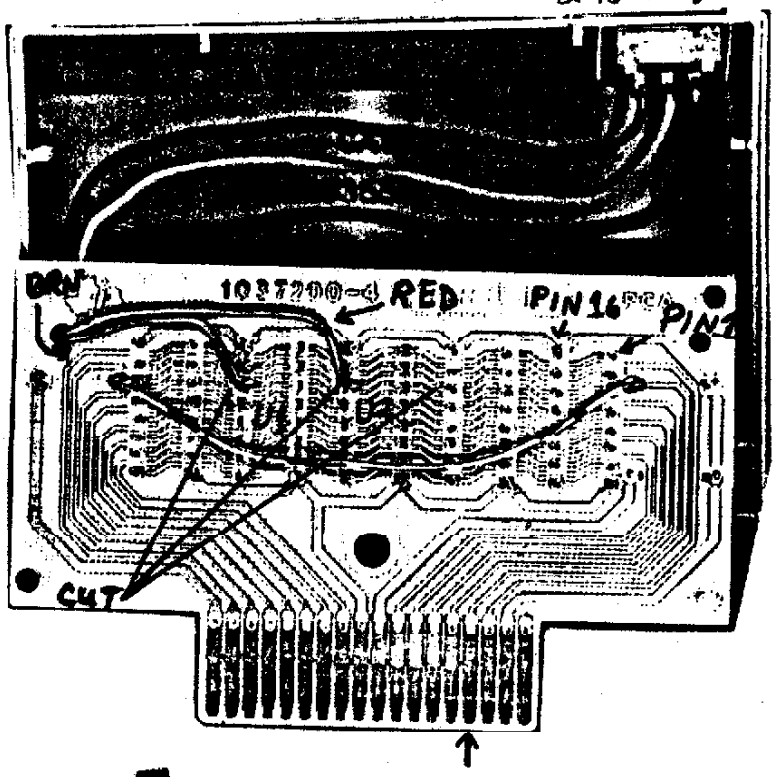
by Jim Ellis

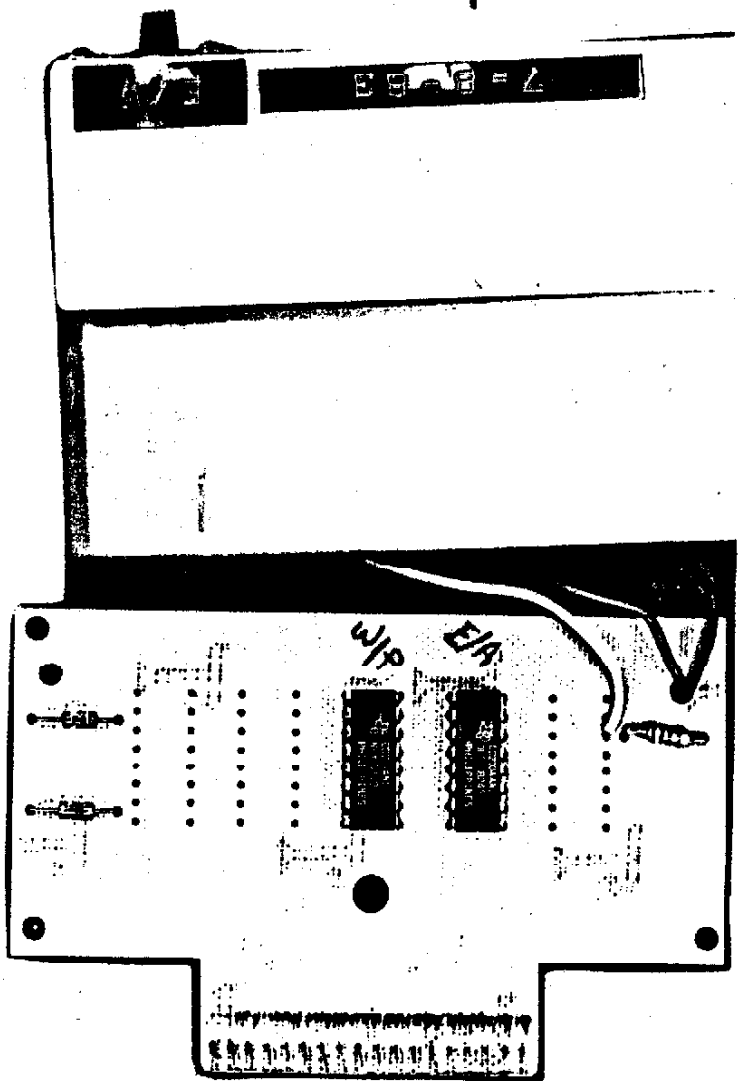
First, let me say that this project is not for the inexperienced. If you don't have the tools or experience or just not sure you want to tackle it, I'm sure you can find someone with the skill to do it for you at a nominal fee. You can have your own cartridge expander or how to make some modules serve double purpose. I recently got tired of swapping my E/A and TI-writer modules. Even tho' I had an expander, I needed to use one more cartridge than it allowed, so I came up with this little modification. I moved my TI-writer GROM into the Editor Assembler module, cut a few lands, added a s.p.s.t. switch and eureka I had it. These boards are laid out with all pins connected together, i. e. pin 1 to pin 1, etc. So you have to cut open the chip select line (pin 14), solder a wire to each pin 14 that you are using, (in my case two), run the wires to a switch, run a wire from the common terminal of the switch to the reset resistor on the card and its done. Pin 14 originally goes to the reset resistor. Also, you must add a jumper from the reset resistor to pin 29 (-5V), see figure. You need to make (n+1) cuts on the line connecting pins 14 together. E.g. two chips require 3 cuts. Oh, yes, the switch is Radio Shack 275-407, which comes two in a pkg., priced @ 79 cents. If you have any questions leave mail in box 12 on the HUGbbs or call me at 831-5791. If you read diagrams, I included a simple schematic.



With the addition of two more switches it could be made to switch four different programs. Since there are five openings on the board, you could have two-one GROM and one-three GROM programs all in one module. How about that? TI fans!! Later.....

Radio Shack
275-407





P R BASE, A REVIEW
by Joe Nuvolini

Editor's Note: The following article comes courtesy of the November, 1985 issue of the "Front Ranger", Newsletter of the Front Range 99'ers of Colorado Springs, Colorado.

This month we're going to take a look at Bill Warren's new database program, PRBASE. Bill is the Secretary of the Rocky Mountain 99'ers and lives in Aurora, CO. PRBASE is being distributed under the Freeware concept and Bill is asking only $10 for the program, an exceptional buy!

What is PRBASE, you ask? Well it is a disk-based fast access database management system written in assembly language. The program currently loads from the Editor Assembler, but Bill is working on an Extended Basic loader. The first thing you do upon receiving the program is to print the PRBASE:DOC file. There is a short file called HOWPRINT that tells you how to do it but it is printed using option 4 of the Editor Assembler. Make sure you have plenty of time, because it takes a while to print, but is an excellent product. NOW READ IT!! Once that is done, you are ready to start. Load the file called CREATE. Good, now you can design an input screen using a maximum of 255 characters divided up into a maximum of 32 fields. Note that if you set off the actual input areas of the screen with brackets, everything you input will be upper case regardless of the alpha lock position. If you set them off with braces, you can enter either upper or lower case. You can lay out the input screen any way you like and even add graphic borders, if you like. One note of caution here, if in the create mode you press FCTN 3, ERASE, while designing your input screen, it erases the entire screen, not just the line you are on. The mailing labels and reports formats are also generated here, and can be up to 10 lines long. The files generated by PRBASE are DISPLAY VARIABLE 80 and are compatible with TI-Writer and the Editor Assembler. Once designed, you save the input screen, label report formats on the disk that will contain your data. The first 10 sectors of your data disk will contain the information you design with the CREATE file. The remaining 350 sectors are reserved to store your data, one input screen per sector, or 350 entries. Bill is planning on increasing that number for those with double-sided capability. Now you're ready to use the PRBASE program itself. Load it and enter your datafils name, and voila, your input screen is displayed. Note that when you first access your data disk, the program will build an index on the first field in your input screen. Therefore, you should place your most significant piece of data at this location. You can change the index field by selecting I, moving the cursor to the desired field and pressing FCTN 6, PROC'D. You must do this if you plan to sort S on that field. Once indexed you can view the contents by entering V and holding down the space bar. The contents will display across the bottom of the screen.

Now let's look at the other functions available in this mode. I guess the first to talk about is H. When you press H you get a help screen that tells you what happens when you press the other keys. If you want to add records press A and the program will display the first empty record starting with the currently displayed record. It's always good therefore, to first press N, screen number, and enter 1 to return to the first screen before selecting A. Now input your data and it will be saved to your data disk, after you complete each screen. If you want to go back and edit a screen you have already finished you must first find it, so let's look at the three search options.

The first is global search, G. Here you can input a string up to 19 characters long and the program will search for the string in all the fields on all the screens. It takes about 90 seconds to search the entire disk. On all the searches pressing FCTN 6 after finding the first occurrence will continue the search. If you press FCTN 7 the current record displayed will be printed to the currently selected output device using the last selected report format. Pressing FCTN 8 will print all of the occurrences of the string selected. The F search will find a string, but will search only one field in all the screens, obviously faster. The last is the U or use index search. This search will find any record on the disk in less than one second. You must use a string that is in the currently index field. Now that you have found your record you

can edit it by pressing E and moving the the cursor over the field you want to edit with the FCTN arrow keys. Enter D if you want to delete a record. The program will ask you if you are sure before deleting the record displayed. The L command will print your mailing labels while R will give you a menu of your report formats to select from. The labels currently print one across but will be changed to offer other options in a future version. The O command allows you to re-select your data output drive # and your output device, i.e., PIO/1, etc. The P command will print the current screen to the output device selected. T toggles the sound on and off in case the kids are asleep and you are inputting data. The B command will allow you to "boot up" another data disk without reloading the PRBASE program. Q stands for quit and after selected you will be asked if you are sure. Answer Y and you are gone!

I saved the C or copy command for last as it needs some explanation. The C command will make a copy of your data disk, if you want one. However there is no copy program on the disk as you receive it. You may put your own on such as Masscopy. If you have this program, you should put MASSCOPY, MC3, and MCINST on your PRBASE disk renaming the MASSCOPY file, COPY.

So much for the action commands. When using the PRBASE program the following are also available:

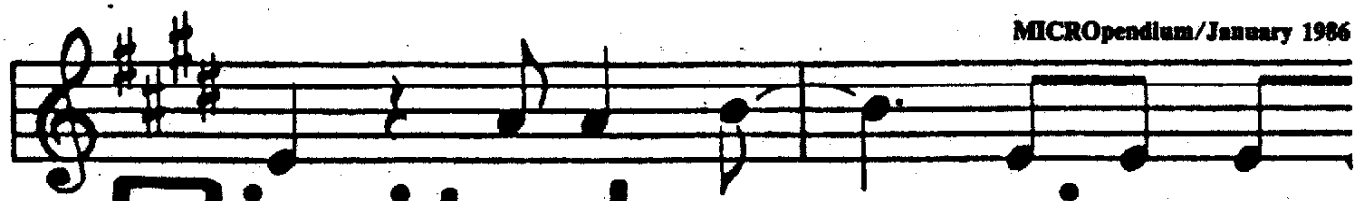
FCTN X Scroll to next screen
FCTN E Scroll to previous screen
FCTN D Next alphabetical screen
FCTN S Previous alphabetical screen
CTRL X Rapid scroll screen 1-350
CTRL E Rapid scroll screen 350-1
CTRL D First alphabetical screen

I've told you all of the good things, now there are a few negatives. The problems listed here are really only inconveniences . They include the lack of two-by labels, the lack of a skip over perf when printing long lists, and the fact that setting up the reports and mailing labels is a bit tedious, and confusing. I have had several calls on this, and as apparently the only one that I know of in our group that has been successful in getting them all to work. The skip over perf can be overcome by printing only fifty names at a time. Some of these things will be corrected in future versions while others will not.

This may seem a bit longwinded, but believe me, I have only touched the surface. The possibilities available for locating data and formatting reports are enormous. Whether you use it for keeping track of club information, as we are doing in our Users Group, or just for your own personal mailing list you will find this program an excellent buy at $10. Comparable commercial database programs currently available for the TI on the market run many times the price of this fine program. If you want to get a copy write to:

William M Warren
2373 Ironton St
Aurora, CO 80010

If you are from the Colorado Springs area tell Bill that you will get your updates from me and he will provide me with a list of those who have paid when he sends me an update. The list as of this writing includes myself and Cliff Swank. This will save you all the cost of sending mailers back and forth each time there is an update. I will contact each of you and make arrangements to get the updated version to you. This one is a winner. Between Dave Vaughan (BITMAC), Tom Frerichs and Mike Holmes (MA/Talk), and Bill Warren (PRBASE), the Rocky Mountain 99'ers have a wealth of programming talent. It is unlikely that three such fine programs have come out of any other single club in America. And get ready for a new BBS program Tom and Mike have in the oven. Look for a review of this one here in the near future.



Digital music

Digital Music by Stephen D. Peacock of Jacksonville, Florida, is designed for would-be computer composers and others who would like to use their consoles as musical instruments.

Digital Music consists of several programs that allow the user to create and play musical pieces. The program allows the user to test his compositions as they are created and to recall them as often as necessary to perfect them. It requires Extended BASIC and a disk system.

The six programs included with Digital Music are called HELP, LOAD, MAKE, PIANO, STAFF and WORK. Copy them from the listings published here and save them to disk under these names. The programs called MAKE and WORK are identical so save it twice under the names MAKE and WORK.

The main menu consists of five options:

1. Instructions
2. To Play Music
3. To Compose A New Song
4. To Add to "Work" File
5. Print Numbers On Staff

The HELP program consists of instructions to operate the program. Much of this text is taken from the HELP program. Option 1 loads this program.

The second option controls the program called PIANO. This program controls the playing of the music. PIANO corresponds to Option 2.

The MAKE program corresponds to Option 3 and is used to compose music.

Option 4 also accesses the MAKE program. Both Option 3 and 4 utilize a fourth program called WORK, which is used to hold data while you are creating and testing a song.

PIANO

When PIANO is run you will see a picture of a piano with a message to "PLEASE WAIT, I AM STUDYING MY MUSIC." This pause is needed to calculate the values of several arrays. When these values are computed, a menu will appear. This menu consists of the following options:

AGAIN--------1
NEW----------2
END/COMPOSE--3
INDEX--------4

To play a song select NEW. You will be asked for the title of a song that has been saved. The prompt "DSK1." will appear. Just type the name of a song and press enter. It will run and play. If you enter an incorrect name you will get the message "DEVICE CAN NOT BE ACCESSED" and you may try again.

To play the same song again, select AGAIN. The song will be repeated.

COMPOSING

To compose a song you must enter the total number of notes you will use in a DATA statement, as well as the duration you want the notes to be played and numbers to correspond to the notes. Digital Music requires three numbers for each note (which are played in chord-like fashion). If you want to have a series of six notes or chords assigned to a single DATA statement the first entry in the DATA statement would be the number 6, followed by a comma, of course. Then you would enter the duration of the first note or chord, followed by a comma, and the three numbers corresponding to the three notes that are required by Digital Music. Here is an example:

490 DATA 3,300,20,20,20,300
,2,21,25,300,3,22,26

Three notes would be played for 300 milliseconds each, consisting of sounds created by the numbers 20, 20 and 20; 2, 21 and 25; and 3, 22 and 26.

You may string as many of these combinations together as you can per DATA statement, just make sure that the first number is equal to the total number of chord or note sequences. Of course, you may use as many DATA statements as it takes to compose your song.

Data is entered using the MAKE program, which can be run from the main menu, the start of this program or by running DSK1.MAKE from a cold start.

Each note corresponds to a position on a musical staff. The lowest is C, which is equal to 110 hertz. C is given the number 1. The next note, C sharp, is 2, with D being 3. D sharp/E flat is 4. This is continued for a total of 45 half steps, up to the G sharp above the treble staff.

To enter a sharp give the next higher number. For example, if F is equal to 6 then F sharp is equal to 7. A flat uses the next lower number. For example: if B is equal to 12 then B flat is equal to 11. To play a rest enter the number 46. This has been assigned a frequency of 44000 hertz.

The duration of the note is entered in milliseconds; 300 is equal to 300 milliseconds.

When the compose option is selected the file DSK1.MAKE will run. You will see the prompt RUN/TEST/BREAK/SAVE. These options are activated by entering the first letter of the one you want to use—R, T, B or S. It is here that the information about DATA statements above is put to work. You would select "B" when writing a song. This will BREAK the program. You will be prompted to start entering data at program line 490. Simply enter NUM 490 and you can start entering musical data.

After the data has been entered, enter RUN. When the prompt line reappears press "T" for TEST. This will read your data and print it to the screen. You will be able to see if you have entered the data properly. If not, it will inform you of your errors. Hold any key to stop the listing. Release the key to restart.

After the data file has been tested select "S" for SAVE. The program will end and prompt you to save the file as DSK1.WORK. If it's saved in this way it can be run from the main program or from the start of the make program. The DATA statements will be intact and you can add to or change them. The file DSK1.MAKE will still

COLOR CHANGES IN THE EDITOR/ASSEMBLER

MICROpendium/August 1985

The essential problem in trying to change the screen or character color in the EDIT program is that absolutely every byte of memory except that actually used by the program itself may be used to store data that has been written. The subprogram needed is actually rather short and simple—but where to put it?

It turns out that TI always puts a line of visible ASCII characters near the beginning of its programs that declare the copyright, etc. These bytes are never actually used, and therefore can be replaced. There are 42 bytes available (16 words of memory)—that's it! The unassembled version follows (you'll never actually use it) and is basically a VWTR to change VDP register 7.

```
AORG >22B2    TWO WORDS TO BE REPLACED, TO BRANCH OUT OF PROGRAM
     BL       @@>2020        THE WORDS REPLACED ARE >C020,>FFD8
AORG >201E
MOV  0,@@>201E  TO SAVE REGISTER 0 WHICH NEEDS TO BE USED
LI   0,>8717  FIRST NYBBLE MUST BE 8, 2ND IS THE REGISTER TO BE CHANGED
***             3RD IS FOREGROUND COLOR,4TH IS BACKGROUND(BLACK ON CYAN)
SWPB 0        NEED TO WRITE LSB FIRST
MOVB 0,@@>8C02  VDP WRITE ADDRESS(AUTO INCREMENTS)
SWPB 0
MOVB 0,@@>8C02  THIS COMPLETES CHANGING VDP REGISTER 7
MOV  @@>201E,0  PRESERVES THE PREVIOUS WS REGISTER 0
MOV  @@>FFD8,0  THESE ARE THE BYTES REPLACED BY THE BL INSTRUCTION
RT              RETURN TO THE ORIGINAL PROGRAM
```

The following instructions apply to using TI-Forth or Disk Fixer to change the necessary bytes. First make a copy of the EDIT1 program onto an empty disk. This is necessary so that you will know exactly where the program is located, namely starting in sector >22 (SCR #8 in Forth). Now get the FORTH DUMP routine into memory and EMPTY-BUFFERS. Type HEX 9 BLOCK and 20CA 4 DUMP. You should see 22B2 C020. (In Disk Fixer this is Sector >24, address 00B8.) Replace these two words with 06A0 2020.

In Forth use the following method: first,type in the replacement words in reverse order, remembering the spaces in between, then press enter (there is room for 16 per entry). Second type SP@ addr bytes CMOVE, where addr is the first address the words are to be moved to, and bytes is twice the number of words. Third, check your results with DUMP. Then UPDATE FLUSH. Now check Sector >22 in Disk Fixer, address 0024-0043 (EMPTY-BUFFERS 8 BLOCK 2236 20 DUMP in Forth). You should see 2843 2920 434F 5059 etc. These 16 words should be replaced by:

2020 C800 201E 0200 8717 06C0 D800 8C02
06C0 D800 8C02 C020 201E C020 FFD8 045B

When you have finished this, type UPDATE FLUSH (in Forth) and you are done! Note that you can of course choose any other colors besides black on cyan by changing the LSB of the word at 223E above (002C in Disk Fixer) to the colors of your choice. See your manual for codes.

Be sure you have a backup copy, which you should anyway. I don't want to be responsible if you lose your only one.

NEWSLETTER GLEANINGS

by Jim Ellis

I've heard of brown from the sun, now I've heard of the Brown Disc Co. They list the prices as $9.99 and $12.99 for box/10, SS/DD and DS/DD, respectively, including shipping but not tax. They guarantee them for YOUR lifetime, if one should ever fail, they will send you two (2) of the same kind in replacement. For info they also have an 800 number or use it for orders.

Moving right along, next for the TI user, a bumper sticker that says "I'm a 99/4A BOOSTER, I love my TI-99/4A, We eat apples for lunch", plus it has some graphics, have we come into our own or what? They sell for $1.50 in quantities of one or 3/$4.

Next, from Pilgrims' Pride, a new product for your consideration. At this writing info is sketchy, but it is all I have. It allows you to load almost any cartridge that was made for the TI onto a disk and run it through the module, expands your system by 8K, provides an Editor/Assembler loader pkg: load and run, run pgm files, print pgm files. It requires 32K minimum, has menu for controlling up to seven pgms in memory at once. No name given for it, but the price is to be $59.95 - $69.95, with a hopeful delivery of Mar 1. They also have several items on sale in their flyer.

As usual if you require any additional information see Pam or myself.

DIGITAL MUSIC—

be available to create a different song. Please note that only one WORK file may be run from this system. If you want to work on more than one song at a time, save them under other names. You will then need to run that file from a cold start.

After you have tested and saved your data, select "R" for run. The PIANO program will then run, and you may listen to your new composition.

To see all of the songs that you have saved to disk, select 4 for INDEX. Every song that has been saved with the MAKE program will be listed. Press enter to step through all of the titles. At any time you may press "Q" to quit, and return to the menu.

That should be enough to get you started on the road to composing with Digital Music.

EDITOR'S NOTE: Due to space limitations, the six programs which came with this article will be available in the HUGger Library at this upcoming meeting.

TIPS FROM THE TIGERCUB

#28

Copyright 1985

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

NUTS & BOLTS DISK No. 2 is now ready, and I think it's better than the first one. It contains 188 utility subprograms in merge format, including many new character fonts and screen display routines as well as 2-dimensional array sorts, variable line numbers in GOSUB, GOTO and RESTORE, on-screen editing and much, much more. The price is $19.95 postpaid, or you can order both Nuts & Bolts disks for $37 ppd.

And I have put together 18 different collection disks each containing 5 or 6 of my catalog programs for just $12 postpaid. The programs on each disk are all of the same category, and I have filled up the rest of the disk with public domain programs of the same category, as a bonus.
I want to make it very plain that I am NOT - repeat, NOT - selling public domain programs! My own programs on these disks are offered at a great discount and the public domain programs are just thrown in for free! Together with this issue of the Tips I am mailing to each user's group a copy of my catalog #6 with an added page describing these new offerings, and a rebate offer to user's groups.

My catalog will be sent to individuals for $1, which is deductable from your first order. If you already have my catalog #6, the added page will be sent to you free on request.

My full disk collections will now be available to bona-fide retailers at standard wholesale prices. Inquiries on your letterhead are invited.

And so, on to old business. Yes, I know that RESequencing a program does not resequence references to line numbers in REMs. I just forgot! In line 278 of the Menu Loader in Tips #27, the reference should be to lines 288 and 298, of course.
While programming the file reader in that menu loader, I ran into a peculiarity of the TI-99/4A that surprised most of the expert programmers whom I called for help. When you "read blind" you must read everything as a string, because attempting to read a string as numeric will crash the program. This is no problem with DISPLAY files - but when I tried it with INTERNAL files, I got the strangest garbage! My solution (not quite foolproof) was to identify a record as numeric if it was 8 bytes long and contained an ASCII out of printable range, and then RESTORE the file, read back to that point and re-read it as numeric. Not very efficient!
The following routine will save a numeric input in an internal file, read it back out as a string, show you the way it was saved, and then attempt to translate it back to numeric. It works for positive and negative integers or non-integers of not less than -99, but not for less than that.

```
100 INPUT X :: OPEN #1:"DSK1
.TEST",INTERNAL,OUTPUT :: PR
INT #1:X :: CLOSE #1
110 OPEN #1:"DSK1.TEST",INTE
RNAL,INPUT :: INPUT #1:A$ ::
 PRINT A$ :: CLOSE #1
120 FOR J=1 TO 8 :: PRINT AS
C(SEG$(A$,J,1));:: NEXT J
130 FOR J=1 TO 8 :: A(J)=ASC
(SEG$(A$,J,1)):: NEXT J
140 X=A(1)-63 :: IF X<73 THE
N 150
142 X=192-A(1):: N$="-" :: F
OR J=2 TO X+1 :: N$=N$&STR$(
256-A(J)):: NEXT J :: GOTO 1
60
150 FOR J=2 TO X+1 :: N$=N$&
STR$(A(J)):: NEXT J
160 IF A(J)<>0 THEN N$=N$&".
"&STR$(A(J))
170 J=J+1 :: IF A(J)<>0 THEN
 N$=N$&STR$(A(J)):: GOTO 170
180 N=VAL(N$):: N$="" :: PRI
NT N :: GOTO 100
```

So, here is another Tigercub Challenge! Can you fix it? Let's HEAR from you this time!

Another problem that I ran into was in recovering from an I/O error. When ON ERROR is used to prevent crashing on such an error, the file is "ajar" - you can't close it and you can't open it. My solution was to simply RUN the program again - and this will show you how the pre-scan speeds that up. Since then, I have learned of three other ways. The method described in the Sydney (Australia) newsletter is a bit complicated, but Irwin Hott gave me a simple solution - just increment the file number! Works fine if you don't increment it into the number of another open file on the disk. Chuck Grimes gave me an even better way - open and close anything else, even "PIO"! Example -

```
100 ON ERROR 110 :: OPEN #1:
"DSK1.TEST",OUTPUT :: PRINT
"CONTINUE PROGRAM" :: END
110 OPEN #1:"PIO" :: CLOSE #
1 :: PRINT "I/O ERROR":"CHEC
K DISK AND DRIVE":"THEN PRES
S ANY KEY" :: ON ERROR STOP
120 CALL KEY(0,K,S):: IF S=0
 THEN 120 ELSE 100
```

There is a reason for that ON ERROR STOP, and it's why I don't use ON ERROR if I can avoid it. When an error occurs, the program goes to the line number specified by the last open ON ERROR statement, takes whatever action is directed by that line, and RETURNs as directed. If the error was not one that you expected to happpen, the results can be very confusing!
For that reason, when you set out to modify a program, the first thing you should do is delete, temporarily, all the ON ERROR statements. The next thing you should do, if the program has a routine to turn off the pre-scan, is to disable that. Otherwise, you will be driven crazy by invalid SYNTAX ERROR messages and other strange happenings.
The third thing you should do is to make a list of all the lines that a GOTO or GOSUB goes to, so you don't delete or change them. And here is a program to do just that for you -

```
100 !GO-SEARCH by Jim Peters
on searches a MERGE format f
ile, finds all line numbers
containing a jump, sorts int
o "to" line number sequence,
110 !prints "to" line number
, statement (GO, GOTO or GOS
UB) and "from" line number
120 DIM C(200):: A=1 :: GO$(
1)="GO" :: GO$(2)="GOTO" ::
GO$(3)="GOSUB"
130 INPUT "FILENAME? DSK1.":
F$
140 OPEN #1:"DSK1."&F$,INPUT
 ,VARIABLE 163 :: OPEN #2:"P
IO"
150 LINPUT #1:A$
160 IF POS(A$,CHR$(133),1)=0
 AND POS(A$,CHR$(134),1)=0 A
ND POS(A$,CHR$(135),1)=0 THE
N 210
170 LN=ASC(SEG$(A$,1,1))*256
+ASC(SEG$(A$,2,1)):: T=133 :
: P=1
180 G$=CHR$(T):: X=POS(A$,G$
```

```
,P):: IF X=0 THEN 200 :: LREF=ASC(SEG$(A$,X+2,1))*256+ASC(SEG$(A$,X+3,1))!:: PRINT #2:LN;GO$(T-132);LREF :: P=X+1 :: GOTO 180
190 C$=STR$(LREF)&","&STR$(LN)&STR$(T-132):: C(A)=VAL(C$):: A=A+1 :: P=X+1 :: GOTO 180
200 IF G$=CHR$(135)THEN 210 :: T=T+1 :: P=1 :: GOTO 180
210 IF EOF(1)THEN CLOSE #1 :: GOTO 220 :: ELSE 150
220 A=A-1 :: CALL LONGSHELLN(A,C())
230 FOR J=1 TO A :: A$=STR$(C(J)):: X=POS(A$,",",1):: Y=VAL(SEG$(A$,LEN(A$),1)):: A$=SEG$(A$,1,LEN(A$)-1)
240 PRINT #2:SEG$(A$,1,X-1);TAB(7);GO$(Y);" FROM ";TAB(21);SEG$(A$,X+1,LEN(A$)):: NEXT J
250 SUB LONGSHELLN(N,NN())
260 D=N
270 D=INT(D/3)+1 :: FOR I=1 TO N-D :: IF NN(I)<=NN(I+D)THEN 300 :: T=NN(I+D):: J=I
280 NN(J+D)=NN(J):: J=J-D :: IF J<1 THEN 290 :: IF T<NN(J)THEN 280
290 NN(J+D)=T
300 NEXT I
310 IF D>1 THEN 270
320 SUBEND
```

According to the User's Reference Guide that came with your computer, if you open a file without specifying INPUT, OUTPUT, UPDATE or APPEND, the computer will assume the UPDATE mode as the default and "UPDATE files may be both read and written. The usual processing is to read a record, change it in some way, and then write the altered record back out on the file." This is a very dangerous bit of misinformation! It is true only if you are using RELATIVE files with the REC clause. In any other case, the first record you write to the file will become the record FOLLOWING the last record you read, and it will also become the LAST record in the file - any records beyond that point will be lost!

The moral of the story - get in the habit of NEVER opening a file without specifying the mode. The only way to update a sequential file is to read it ALL into an array, update it, and then write it back to the file.

I reviewed hundreds of programs, in my PD library of about 2600, in order to select some of the best to fill up the collection disks. Often they needed only a few minor changes to greatly improve them.

One frequent flaw was in interpreting the status of CALL KEY. The User's Reference Guide says that a status variable of -1 means that "the same key was pressed during the performance of CALL KEY as was pressed during the previous performance." This is misleading. It actually means that the same key is STILL BEING pressed. Try this -

```
100 DISPLAY AT(12,1)ERASE ALL:"TYPE YOUR NAME" :: R=12 :: C=3
110 CALL KEY(0,K,S):: IF S=0 THEN 110 :: DISPLAY AT(R,C):CHR$(K):: C=C+1 :: GOTO 110
```

Difficult to type without unwanted repetition of letters? Now try changing the S=0 to S<1 !

IF S<1 (if S is less than 1) means that if no key is pressed (S=0) or if the same key is still being held down (S=-1) then CALL KEY again.

Another frequent flaw is

```
INPUT "WANT TO PLAY AGAIN? ":Q$ :: IF Q$<>"Y" THEN END
```

- or, more professionally programmed, IF SEG$(Q$,1,1)<>"Y" THEN....,

which will accept either "Y" or "YES" as a reply. The problem is still that this question is often asked at the end of a joystick game, for which the Alpha Lock will be unlocked - and a response of a lower case "y" then terminates the program! One solution is to precede the INPUT with a dummy CALL KEY(3,K,S), which will cause any subsequent upper case CALL KEY, INPUT, LINPUT or ACCEPT AT response to be read as lower case until you turn it off with CALL KEY(5,K,S).

Here's one that does nothing except look pretty.

```
100 DISPLAY AT(3,8)ERASE ALL:"COLORSQUARES" :: DISPLAY AT(8,1):"Select option 1, 2 or 3" ! by Jim Peterson, Tigercub Software
110 CALL KEY(0,K,ST):: IF ST=0 OR K<49 OR K>51 THEN 110 :: ON K-48 GOTO 150,120,130
120 FOR CH=38 TO 142 STEP 8 :: CALL CHAR(CH,RPT$("A55A",4)):: NEXT CH :: GOTO 150
130 FOR CH=38 TO 142 STEP 8 :: FOR L=1 TO 4 :: RANDOMIZE :: X$=SEG$("0818243C425A6781C0199A5BDC3DBE7FF",INT(16*RND+1)*2-1,2)
140 B$=B$&X$ :: C$=X$&C$ :: NEXT L :: CALL CHAR(CH,B$&C$):: B$,C$=NUL$ :: NEXT CH
150 CALL CLEAR :: RANDOMIZE :: FOR SET=0-(K>49)TO 14 :: CALL COLOR(SET,SET+2+(K>49),SET+2):: NEXT SET
160 Y=INT(4*RND+3):: R=INT(12*RND+1):: R2=25-R-Y :: C=INT(7*RND+7):: C2=32-C-Y :: IF K=49 THEN X=INT(14*RND+1)*8+22 ELSE X=INT(13*RND+1)*8+38
170 FOR T=R TO R+Y :: CALL HCHAR(T,C,X,Y):: CALL HCHAR(T,C2,X,Y):: NEXT T
180 FOR T=R2 TO R2+Y :: CALL HCHAR(T,C,X,Y):: CALL HCHAR(T,C2,X,Y):: NEXT T :: GOTO 160
```

The asterisk on the Gemini printer looks rather like a bug squashed side- ways, and it was confusing some folks in the condensed print of my newsletter, so I improved it with this -

```
150 PRINT #2:CHR$(27);CHR$(42);CHR$(1);CHR$(42);CHR$(9);CHR$(8);CHR$(34);CHR$(8);CHR$(8);CHR$(62);CHR$(8);CHR$(8);CHR$(34);CHR$(8);
```

And at the same time I improved the slashed zero -

```
140 PRINT #2:CHR$(27);CHR$(42);CHR$(1);CHR$(48);CHR$(8);CHR$(64);CHR$(38);CHR$(96);CHR$(17);CHR$(72);CHR$(5);CHR$(66);CHR$(61);CHR$(8);
```

```
90 !THIS WON'T WORK, WILL IT ?
100 DISPLAY AT(9999,9999)ERASE ALL:SEG$("CAN'T DO THAT!",1,3)&SEG$("CAN'T DO THAT!",6,8)
```

If the Tigercub Math Puzzle in Tips #27 was a bit too tough, these changes will add a couple of easier levels.

```
105 DISPLAY AT(6,1):"Level 1, 2, 3 or 4?" :: ACCEPT AT(6,21)VALIDATE("1234"):L$ :: L=VAL(L$)
106 IF L<3 THEN M$="Insert +, -, or * (multiply)" ELSE M$="Insert +, -, * (multiply) or / (divide)"
110 DISPLAY AT(5,1):M$;" between the digits":" to equal the total": :"Type Q to give up"
120 ! **DELETED LINE **
130 DISPLAY AT(12,1):" " :: T,I=INT(9*RND+1):: M$=STR$(I):: Z$=M$&" "
140 FOR J=1 TO 4 :: Y(J)=INT(9*RND+1):: Q=3+ABS(L>2):: Z=INT(Q*RND+1):: ON Z GOSUB 240,250,260,270 :: Z$=Z$&STR$(Y(J))&" " :: NEXT J
150 IF L/2<>INT(L/2)AND T<>INT(T)THEN 130 :: Z$=Z$&"="&STR$(T)
```

MEMORY FULL

Jim Peterson

HOOSIER USERS GROUP DIRECTORY

HOOSIER USERS GROUP OFFICERS

President.....Steve Sims 631-7255
Vice-President.....Bill Lucid 291-3995
Secretary.....Barb Uhrig 357-8268
Treasurer.....Bill Jones

COMMITTEE CHAIRPERSONS

Regional Centers:
South.....Dennis Sherfy 881-5918
West Indiana..Vic Kelson 812-234-5533

Documents.....Don Donlan 882-4544
Membership.....Pam Sims 631-7255
Newsletter.....Pam Sims 631-7255

MONTHLY MEETING LOCATION

Creative Logic
8240 Indy Lane
Indianapolis, IN 46224

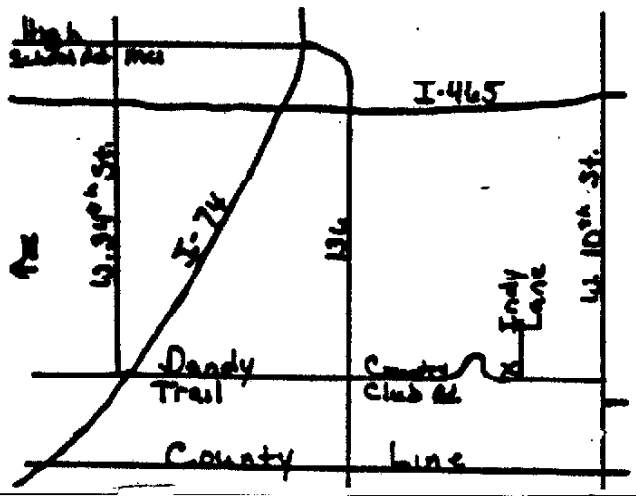
(About 1800 North Country Club Road)

NEWSLETTER EXCHANGE

The Hoosier Users is participating in a Newsletter Exchange program with other TI Users Groups. This offer is made with the understanding that, with proper credit, your Users Group can reprint articles from the Hoosier Users Group Newsletter, and with proper credit, we can reprint articles from other TI Users Groups Newsletters.

PRINTOUTS

Library listings can be ordered for $.25 & a 6x9 self addressed envelope with $.66 postage. The HUGbbs Reference Guide can be ordered for $.50 and a 4x9 self addressed envelope with $.22 postage. Please send orders to our P.O. Box. SORRY, PRINTOUTS WILL BE SENT TO ACTIVE MEMBERS ONLY!



HUGbbs INFORMATION

317-631-994A

The HUGbbs operates on a 24 hour basis.

SPONSOR THE HUGbbs: Any member or retail business can sponsor the HUGbbs. For a $5.00 donation, you get 5 (40 column) lines on the Log-On Title Screen for a week (or for a $10.00 donation, you get 10 (40 column) lines) plus a 24 line by 40 character ad in the Sales option of the File Module. To sponsor the HUGbbs, send a check or money order to our P.O. Box (or turn in at our Monthly Meeting) specifying how many weeks (and how many lines) you want to sponsor, your name (or company name), address, phone, what you want to say, and the week (and an alternate week) you want the ad to appear.*

BACK ISSUES

Back Issues purchased at the monthly meeting is $1.00 each. Mail order price is $1.50 per Newsletter (postage included). Orders will be filled within 3 weeks of receipt by the Documents Committee.

ADVERTISING POLICIES

There will be no charge for advertisements submitted to the HUGger Newsletter by members (for private sale only). Format for the advertisements is 45 characters wide by 10 lines long. The Ad should be typed or hand printed exactly how it is to appear in the Newsletter. Deadline for an ad to appear in next month's Newsletter is the 2nd Saturday of the month.*

For companies who wish to advertise in the HUGger Newsletter, our rates are as follows:

Pre-Printed Inserts (one page) $20.00
One Full Page (one sided) Ad: $25.00
One Half Page Ad: $13.00
One Quarter Page Ad: $7.00

All ads must be in a ready to print condition. Advertisements must be in our P.O. Box before the 2nd Saturday of the month to appear in the following month's Newsletter.*

*NOTE: The Officers of the Hoosier Users Group reserve final approval on all advertisements submitted for the HUGger Newsletter and the HUGbbs. The Officers and the Newsletter committee are not responsible for typographical errors due to illegible advertisements. All proceeds are accepted as donations to the Hoosier Users Group.

FEB. 5 1986

HOOSIER USERS GROUP
P.O. Box 2222
Indianapolis, IN 46206-2222

Forwarding and Address
Correction Requested

Bulk Rate
U.S. Postage
PAID
Indianapolis, IN
Permit No. 6440

TIME DATED
FEB. 5 1986
MATERIAL

APPLICATION FOR MEMBERSHIP

Below you will find an application for membership to the Hoosier Users Group. Active membership entitles you to the Newsletter, up and download on the HUGbbs, attendance and voting rights at regular club meetings, access to the HUGger Library of Programs, special club activities and special guest speakers for one year. Subscribing members will receive the NEWSLETTER only.

Make check or money order payable to **Hoosier Users Group**. Send completed application to:

HOOSIER USERS GROUP
P.O. Box 2222
Indianapolis, IN 46206-2222

(Cut on dotted line)

Check One:

Active Member
New: $20
Renewal: 15

Subscribing Member
New: $10
Renewal: 7.50

Amount Enclosed: $ _____

_____ D _____
S _____ O _____

Name: _____ Today's Date: _____

Address: _____ Apt. # _____

City: _____ State: _____ Zip: _____

Phone: (_____) _____-_____

Interests/Comments: _____